TP1 Réseaux

UJF IMAG

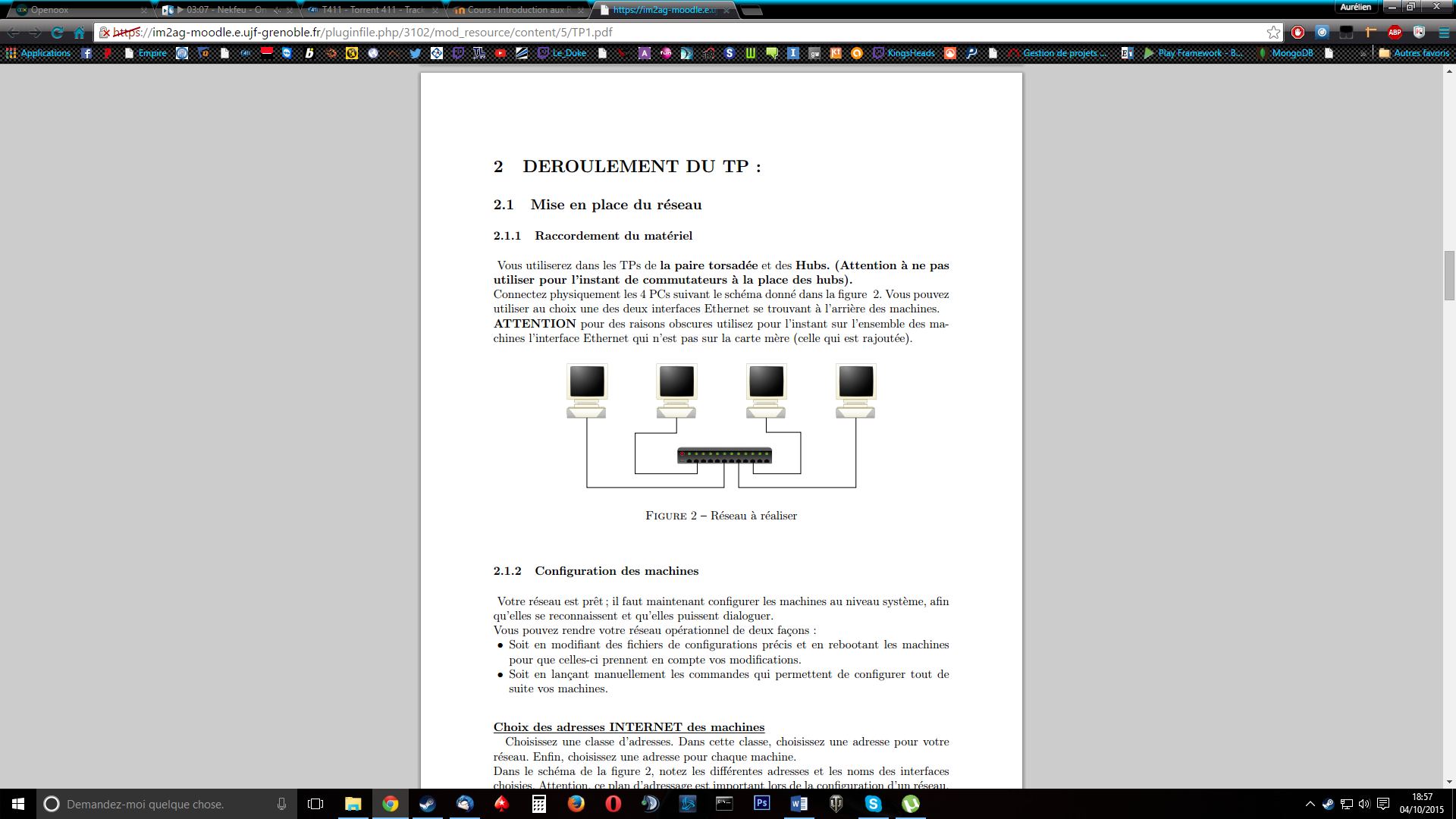
Assemblage et configuration d’un réseau : Observations et mesures

Aurélien Monnet-Paquet – Thomas lerchundi

2015 2016

2. Mise en place du réseau.

Nous avons créée le réseau suivant à l’aide d’un Hub et de paires torsadés.

Nous avons attribués arbitrairement des adresses aux machines, allant de 192.168.0.1 à 192.168.0.4

192.168.0.4

192.168.0.3

192.168.0.2

192.168.0.1

Commande pour changer l’adresse des machines : « ifconfig bge0 192.168.0.1-4 up »

A partir de ce moment, la commande « ping 192.168.0.4 » lancé sur 192.168.0.1 marche ainsi que la commande « telnet –y 192.168.0.2 »

Fonctionnement de ping :

* Une première machine envoie une requête « icmp request » à une seconde machine.
* La machine cible répond à la requête avec un paquet « icmp reply »

Les paquets de type ARP :

Une machine (pc1) d’un réseau doit envoyer un paquet à une autre machine (pc2) du même réseau. Pc1 connait l’adresse IP mais ne connait pas l’adresse MAC de pc2. Pc1 va donc envoyer un paquet de type ARP en broadcast pour identifier la machine cible (pc2). Alors quand pc2 va recevoir le paquet de type ARP, elle (et uniquement elle) va renvoyer une réponse sur le réseau. Ainsi, pc1 connait maintenant l’adresse MAC associés à l’adresse IP.

**La table ARP** contient les correspondances entre les adresse IP et les adresses MAC des machines sur un réseau local. Ce qui a pour effet de ne pas envoyer une requête ARP à chaque fois qu’une machine a besoin d’envoyer un paquet (par exemple de type ICMP) a une autre machine sur le réseau.

2.3. Observation du protocole CSMA/CD

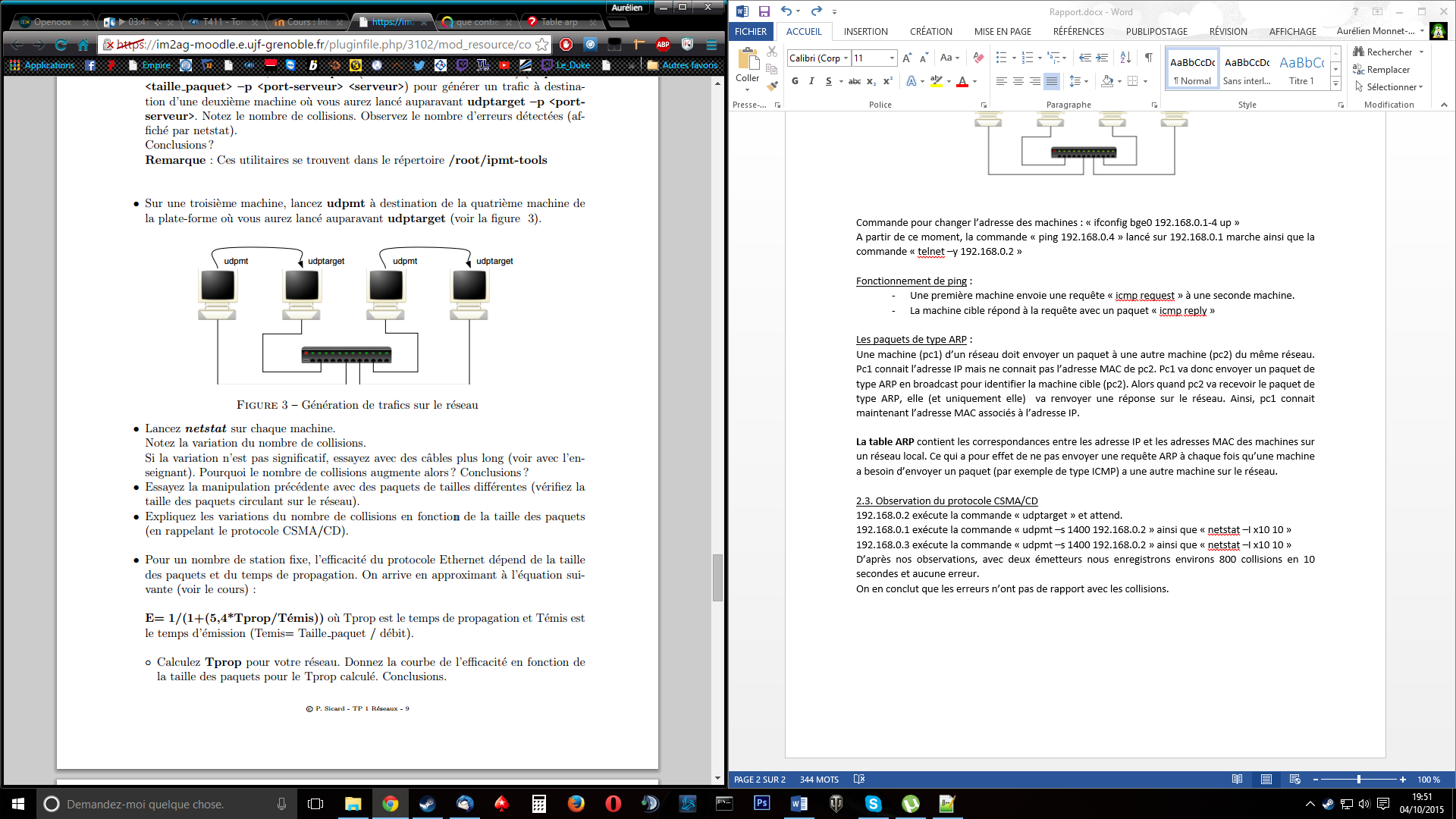
192.168.0.2 exécute la commande « udptarget » et attend.

192.168.0.1 exécute la commande « udpmt –s 1400 192.168.0.2 » ainsi que « netstat –I x10 10 »

192.168.0.3 exécute la commande « udpmt –s 1400 192.168.0.2 » ainsi que « netstat –I x10 10 »

D’après nos observations, avec deux émetteurs nous enregistrons environs 800 collisions en 10 secondes et aucune erreur.

On en conclut que les erreurs n’ont pas de rapport avec les collisions.



Je n’ai pas les résultats de cette expérience.

Le protocole CSMA/CD consiste à écouter le réseau avant d’émettre. Si personne n’est en train d’émettre alors la machine qui écoute peut envoyer des paquets sur le réseau. Plus la taille des paquets est petite et plus il y a de collisions. Plus la taille des paquets est petite et plus la carte peut envoyer de paquet dans un même laps de temps. Ce qui a pour effet d’augmenter la probabilité de collisions sur le réseau.

Tu dois avoir le résultat pour Tprop.

**Pourquoi le partage est à peu près équitable ? Et quel Protocol gère cette équité ?**

2.4.1 Mesure du débit applicatif

Je n’ai pas données là-dessus…

2.4.2 Mesure du débit dans le cas de plusieurs trafics

Les deux machines avec « udptarget » reçoivent un débit moyen de 3 500 kbits/sec.

Comparaison avec 2.4.1 ?

2.4.3 Mesure de latence

Résultat du *ping* : 0.5 ms soit une *latence* de 0.250 ms.

Formule pour évaluer le temps dans les couches protocolaires ?

Résultat du *ping* (taille 1000) : 2 ms soit une *latence* de 1 ms.

Conclusion : Le temps passer dans les couches protocolaire est quasiment le même avec des taille de paquets complètement différentes.

2.4.4 Mesures en utilisant des commutateurs (switch)

Pas de données …

Les switchs de type « store and forward » transmettent en mode différé. C’est-à-dire que le commutateur met en tampon et réalise une opération de contrôle sur chaque trame avant de l’envoyer. Ce qui a pour effet de retardé la transmission mais cela garantit l’intégrité de la trame et évite les erreurs.